Automatic Keyword Extraction from Historical Document Images

Kengo Terasawa, Takeshi Nagasaki, and Toshio Kawashima

School of Systems Information Science, Future University-Hakodate,
116–2 Kamedanakano-cho, Hakodate-shi, Hokkaido, 041–8655, Japan
{g3103004,nagasaki,kawasima}@fun.ac.jp

Abstract. This paper presents an automatic keyword extraction method from historical document images. The proposed method is language independent because it is purely appearance based, where neither lexical information nor any other statistical language models are required. Moreover, since it does not need word segmentation, it can be applied to Eastern languages where they do not put clear spacing between words. The first half of the paper describes the algorithm to retrieve document image regions which have similar appearance to the given query image. The algorithm was evaluated in recall-precision manner, and showed its performance of over 80–90% average precision. The second half of the paper describes the keyword extraction method which works even if no query word is explicitly specified. Since the computational cost was reduced by the efficient pruning techniques, the system could extract keywords successfully from relatively large documents.

1 Introduction

In this paper an automatic keyword extraction method from historical document images is described. Since the expanding usage of digital archives, we are facing an extreme amount of historical document archives. To make beneficial use of these treasures, it is quite important to make indices of these document images. However, the difficulty of making indices of historical documents leads the fact that only a few documents with great importance are allowed to have their indices that are made by the hand of experts. Therefore, it is now essential to develop systems for making indices of historical document images automatically. Such systems will extend the capability of digital archives.

The difficulty of extracting keywords from historical documents is caused by some reasons. One is that historical documents are mostly handwritten and sometimes are significantly degraded due to the passage of time, therefore traditional OCR (Optical Character Recognition) techniques cannot be easily applied. Moreover, the shortage of lexicons and written character samples make the problems more difficult. Such difficulty about OCR application increases the importance of text retrieval method without recognition.

The idea of text retrieval without recognition is seen in the work of Manmatha et al. [4], which they called “word spotting”. Rath and Manmatha proposed a

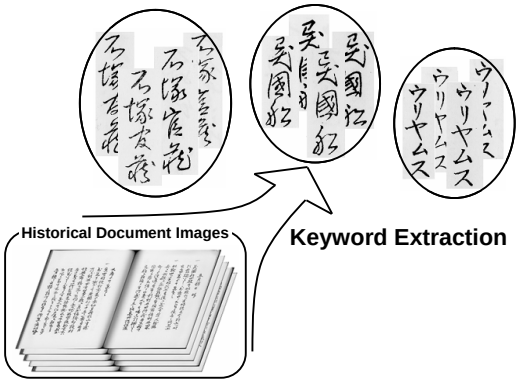


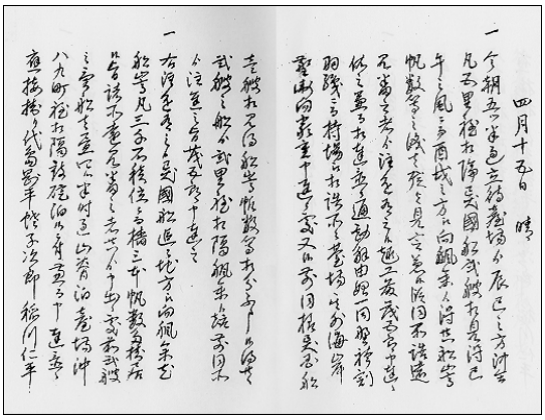
Fig. 1. The purpose of the study

set of features suitable for word image matching [7], and they applied a dynamic time warping method to match the images [8]. Gatos et al. [2] aimed to retrieve texts from historical typewritten Greek documents, where they showed that the use of user's feedback improved the retrieval results. In all of their studies, matching target was a set of segmented words because their works were intended for Western (English or Greek) manuscripts, where word segmentation is possible.

On the other hand, in Eastern languages such as Japanese or Chinese where they do not put clear spacing between words, segmentation into words is practically impossible. A study of word spotting for such a languages was done by Yue Lu and Chew Lim Tan [3], where they developed a method for searching for words in Chinese newspapers. However, it was only applied to machine printed fonts.

The proposed method aims to extract frequently appearing words from historical document images, as displayed in Fig. 1. Although it is mainly intended for handwritten manuscripts of Japanese historical documents, the proposed method is ideally language independent in the sense that it is completely data-driven. It does not need lexical information nor any other statistical language model. Our text retrieval method and keyword extraction method are both based on image matching techniques, i.e. appearance-based. Since we avoid using any lexical or linguistic information, as a natural consequence, the results our method provide will not be a set of true keywords, but a set of frequently appearing words, including not only true keywords but also stopwords or some other meaningless character sequences. However, it is still helpful in making indices of historical documents, because it provides a good list of candidates for keywords.

Our image matching method was inspired by "eigenface" [10, 11] method, which is widely used in the area of face recognition. In this method, face images are compared in reduced dimensional space. We extended the eigenspace method to compare the sequences of images by the use of sliding window technique. The



(a) The first and second pages of scanned images

又左衛門 (165)	石塚官蔵 (25)
ウリヤムス (73)	井上富左右 (25)
稲川仁平 (24)	安間純之進 (14)
平山謙次郎 (18)	工藤茂五郎 (11)
蛭子次郎 (13)	藤原主馬 (10)
勘解由 (14)	異人共 (79)
異国船 (30)	応接所 (26) 亀田濱 (14)
本線江引取候 (21)	発砲いたし候 (14)
別紙應接書 (22)	見廻方當番 (23)
二番入津之異船 (10)	沖ノ口役所 (15)
警固之者 (11)	上陸いたし (17)

(b) The ground truth for keyword (frequency)

Fig. 2. Materials used in the experiment: "Akoku Raishiki"

use of sliding window technique in feature extraction from handwritten document images is also seen in literature, e.g. Marinai et al. [5], Fink and Plötz [1], etc.

The proposed method starts with making ranked list sorted by similarity to a query image, as described in our previous study [9]. Section 2 reviews the method for making ranked list, and besides, its performance is evaluated in recall-precision manner. In Sect. 3, we present its extension to keyword extraction, and its experimental results are shown in Sect. 4. Finally, Sect. 5 concludes the paper and discusses the future work.

1.1 Materials

In this paper several experimental results are shown. In such experiments, the materials used were scanned images of "Akoku Raishiki (The diary of Matsumae Kageyu)" (Fig. 2(a)). It is a historiography written by a Japanese government worker in the mid 19th century, which consists of 182 pages, 1553 lines, and 25148 characters. A perfect transcription of the document was available by grace of our civic library.

For use with retrieval evaluation in Sect. 2, we have manually marked every appearing instances in the images for several keywords. In this marking step, the perfect transcription was of course helpful, but final products were surely handmade.

The ground truth of keyword, which is the main objective to be extracted in Sect. 3, was constructed by n-gram statistics and handmade exclusion. First, character strings with at least 10 frequency and at least 3 character length were extracted by n-gram statistics. This constraint was employed for the reason that shorter character strings are tend to be a function word or stopwords. After that, stopwords, meaningless character strings, and duplications were removed by hand. As a result, we have made a set of keywords consist of 23 words (Fig. 2(b)).

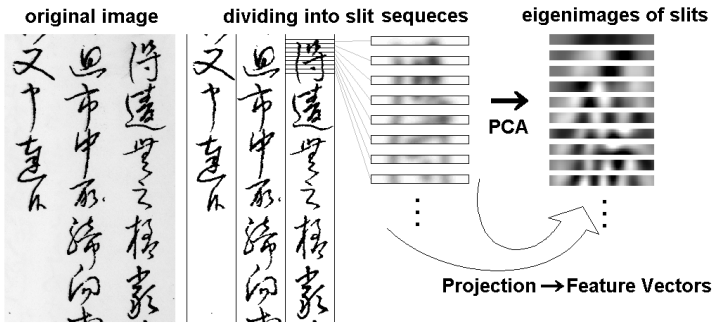


Fig. 3. The original image was transformed into the sequences of features by means of sliding window technique and eigenspace projection

2 Ranked List Making

This section describes the algorithm to make the ranked list sorted by the similarity to a certain given query image. The document image was divided into a sequence of small slit style images, and transformed into a feature space as illustrated in Fig. 3. This method is similar to our previous study [9], but differing in some respects because it includes our recent improvement.

2.1 Preprocessing

Some preprocessing steps needed to be performed before main process. That is, adjustment of image resolution, background removal, line separation, realignment to remove the perturbation of text lines, and Gaussian smoothing. Image resolution and Gaussian smoothing parameter have to be determined with consideration. Our preliminary experiments derived that about 80 pixels resolution per line width and Gaussian parameter $\sigma = 4.0$ gave the finest result.

2.2 Transformation into Slit Sequences

Preprocessed images were then transformed into a sequence of slits. "Slits" mean narrow rectangular windows that scan images along the line axis. The optimal width of the slits (length of the window along the line axis) was also obtained in preliminary experiments as tenth part of the character size: in this case 8 pixel width. Being different to our previous study, cut windows had no overlapping. It is because our recent experiments detected that the dependency on the origins of slit cutting was sufficiently avoided by Gaussian smoothing.

2.3 Eigenspace Projection

Each slit images are transformed into low dimensional descriptors by means of PCA. The covariance matrix of mean-adjusted image vectors was calculated as:

$$C = (\boldsymbol{x}_1 - \boldsymbol{c} \quad \boldsymbol{x}_2 - \boldsymbol{c} \quad \cdots \quad \boldsymbol{x}_m - \boldsymbol{c})(\boldsymbol{x}_1 - \boldsymbol{c} \quad \boldsymbol{x}_2 - \boldsymbol{c} \quad \cdots \quad \boldsymbol{x}_m - \boldsymbol{c})^T, \quad (1)$$

where m represents the number of slit images, $\boldsymbol{x}_i$ represents the image vector of i-th slit, $\boldsymbol{c}$ represents mean image vector. The eigenvectors for the covariance matrix belonging to the d-largest eigenvalues were chosen as $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_d$. These vectors were called principal vectors, and formed the basis of eigenspace.

Each one of the mean-subtracted images were then projected to these basis vectors, and the generated m d-dimensional vectors became a good descriptor of the original image. These low dimensional vectors allowed us to solve matching problems more easily.

2.4 Matching by Slit Feature Descriptors

After the image document was transformed into vector sequences, the remaining problem was matching the vector sequences.

Let $\boldsymbol{y}(t)$ be the sequence of slit features, where t represents slit ID. Let $\boldsymbol{A} = \{\boldsymbol{y}(t) \mid t_0 \leq t \leq t_0 + \tau\}$ be the feature sequences of the query image, and $\boldsymbol{B} = \{\boldsymbol{y}(t) \mid t'_0 \leq t \leq t'_0 + \tau\}$ be the arbitrary feature sequences that has the length same to the query. The matching cost between $\boldsymbol{A}$ and $\boldsymbol{B}$ was defined as

$$D(\boldsymbol{A}, \boldsymbol{B}) = \sum_{0 \leq t \leq \tau} d\left(\boldsymbol{y}(t_0 + t),\ \boldsymbol{y}(t'_0 + t)\right), \tag{2}$$

where $d\left(\boldsymbol{y}(t_0 + t),\ \boldsymbol{y}(t'_0 + t)\right)$ represents the distance between two feature vectors. Computing $D(\boldsymbol{A}, \boldsymbol{B})$ for all possible $\boldsymbol{B}$s, and the feature sequence $\boldsymbol{B}$ which gave the minimum matching cost was selected as the retrieval result. Note that "possible $\boldsymbol{B}$s" were much larger in our Japanese case where segmentation into words is impossible, than English case where segmentation into words is possible. We must consider sequences starting from arbitrary slit ID. The number of the times we have to compute D is the same as the number of slits, which is almost ten times larger than the number of characters.

Here, we must mention the definition of the distance between two feature vectors. Although several ways can be used to define this distance, we employed the most common Euclidean distance, i.e.,

$$d\left(\boldsymbol{y}(t_0 + t),\ \boldsymbol{y}(t'_0 + t)\right) = \sum_i |y_i(t_0 + t) - y_i(t'_0 + t)|^2, \tag{3}$$

where y_i represents the i-th element of vector $\boldsymbol{y}$. In our preliminary experiments we have found that the Euclidean distance gave the best performance especially when used with DTW described in the subsequent section. When DTW was not used, Manhattan distance and Euclidean distance delivered similar performance.

2.5 Dynamic Time Warping

To make the matching algorithm more robust, we applied dynamic time warping (DTW). DTW is a widely used method in the area of speech recognition. If two time series are given, DTW considers every conceivable time correspondence

including non-linear time coordinate transformation, and it outputs the path with minimum matching cost. For our document image retrieval, regarding a slit ID as a time coordinate, sequences of slits can be regarded as a time series.

The time normalized distance between two vector sequences $\boldsymbol{A} = \{\boldsymbol{y}(t) \mid \alpha_1 \leq t \leq \alpha_n\}$ and $\boldsymbol{B} = \{\boldsymbol{y}(t) \mid \beta_1 \leq t \leq \beta_m\}$ was defined as follows:

$$D(\boldsymbol{A}, \boldsymbol{B}) = \min \left[\frac{\sum_{\theta=1}^{k} d\left(\boldsymbol{y}(i_\theta),\ \boldsymbol{y}(j_\theta)\right)}{k} \right], \tag{4}$$

where $(i_1, j_1), \ldots, (i_k, j_k)$ represents the path, satisfying

$$(i_1, j_1) = (\alpha_1, \beta_1) \tag{5}$$

$$(i_k, j_k) = (\alpha_n, \beta_m) \tag{6}$$

$$(i_\theta,\ j_\theta) = \begin{cases} (i_{\theta-1}+1,\ j_{\theta-1}) \\ (i_{\theta-1}+1,\ j_{\theta-1}+1) \\ \text{or } (i_{\theta-1}+1,\ j_{\theta-1}+2), \end{cases} \tag{7}$$

$$1/\alpha\ (i_k - i_1)\ \leq\ j_k - j_1\ \leq\ \alpha(i_k - i_1). \tag{8}$$

where k represents the length of the path, which takes the same value as n in this definition. Equation (5) and (6) represent boundary condition, (7) represents recurrence, and (8) defines global constraint in order to prevent excessive warps, where α is the stretching allowance ratio. In the following experiment, α was set to 1.2, which showed the best performance in our preliminary experiment.

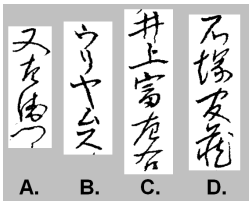
Equation (8) differs from standard DTW constraint such as parallel band or diamond band because such standard constraints are only used when considering matching problem of separated sequences, i.e. starting points and ending points are strictly specified. On the other hand, our problem considers matching problem of continuous sequences, where arbitrary slit may become a starting point or ending point. This type of problem is also dealt in [6], where CDP (Continuous DP) method was introduced. CDP uses sophisticated recurrence equation and prevent warps with more than twice stretching ratio. Equation (8) is similar to CDP, except that it allows arbitrary limitation of maximum stretching ratio.

2.6 Experimental Evaluation

In order to evaluate the performance of the obtained ranked list, we have adopted a recall-precision evaluation, which is widely used in information retrieval researches. Recall is the ratio of the number of correctly retrieved words to the number of total relevant words. Precision is the ratio of correctly retrieved words to the number of retrieved words. Measuring precisions at various recall levels, the recall-precision curve is produced. Average precision is the mean of the precision values obtained after each relevant word to the query has been retrieved.

In this experiment, we have selected four keywords from the whole document, as shown in Table 1. All selected keywords were human name appearing at least 25 times in the document. For each keyword, each appearing instance was used

Table 1. Average-precision for some keywords of "Akoku-Raishiki"



keywords	frequency	average-precision (%)	
		without DTW	with DTW
A. Matazaemon	165	64.12	87.47
B. Uriyamusu	73	71.69	95.23
C. InoueTomizou	25	57.38	92.84
D. IshizukaKanzou	25	58.56	84.20

as a query. The retrieved images were regarded to be correct if the retrieved region and corresponding manually marked region (described in Sect. 1.1) were overlapping sufficiently (in practice, errors less than 20 slits were allowed). The mean of the average precision scores for all queries in each class was calculated for both with-DTW case and without-DTW case, summarized in Table 1.

The result shows that our ranked list was good at its performance, especially when used with DTW.

3 Keyword Extraction

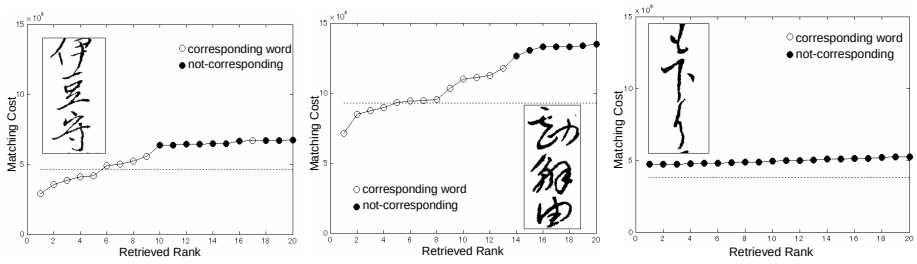
In this section, keyword extraction method is described. Our objective was to extract character strings which appears more than 10 times in the document. To avoid extracting many stopwords, the lower bound of character length in extraction was restricted to 40 slits, which correspond to three or four characters.

First, we introduce a criterion to find out if a certain image region is worthy or not to be selected as a keyword. Then, by applying this criterion to every sub-region in the whole document image, the candidates of keywords were produced. Finally, by clustering the candidates, the set of keywords was constructed.

3.1 Acceptance Check for Each Similarity

In the last section, we have described about the algorithm which gives ranked list sorted by similarity for the given specific image. The next problem is to judge whether this image is keyword or not. This problem has great concern with the judgment of whether the provided similarity is truly coming from correspondence or just only accidental similarity. One idea of making this judgment is to use global thresholding method for matching cost. However, this simple idea ends in failure as illustrated in Fig. 4. In the figure, top 20 ranked matching costs are plotted for three specific images. White circles indicate the valid correspondence, while black circles indicate that similarity was not caused by valid correspondence but by accident. Clearly, the white circles in (b) are higher in position than the black circles in (a) and (c). Therefore, global thresholding method was definitely rejected. Another artifice needed to be employed.

The solution we have found was to normalize distance measure by the energy of image. "Energy of image" means the sum of squared difference of the image



(a) Keyword: "Izunokami" (b) Keyword: "Kageyu" (c) Meaningless character strings

Fig. 4. The matching cost of ranked list for different query images

to a plain white image, i.e.

$$\begin{aligned} \text{Energy of image} &= \sum_{\text{all pixels}} |I_{x,y} - 255|^2 \\ &\approx \sum_{\text{all slits}} d\left(\boldsymbol{y}(t),\ \boldsymbol{w}\right), \end{aligned} \tag{9}$$

where $\boldsymbol{w}$ represents the descriptor into which plane white slit image is transformed. Based on this energy, (4) was normalized as

$$\tilde{D}(\boldsymbol{A}, \boldsymbol{B}) = D(\boldsymbol{A}, \boldsymbol{B}) \ / \ \sum d\left(\boldsymbol{y}(t),\ \boldsymbol{w}\right). \tag{10}$$

In Fig. 4, dashed lines represent the energy of images multiplied by 0.20. The figure indicates that a border dividing white circle and black circle is about proportional to energy of query images. Therefore, the normalization described above seems promising.

3.2 Keyword Candidates Extraction from Whole Document

Based on the normalized distance measure, a criterion to extract keyword was defined. That is, an image region was extracted as a candidate of keyword if its average matching cost of top 10 ranked retrieval did not exceed a certain threshold value (Remember that our objective keyword is frequently appearing words at least 10 times). The threshold value we have employed was 0.20, as displayed in Fig. 4. By applying this criterion to every sub-region in the whole document image, we may extract the candidates of keywords.

Theoretically, the above discussion is enough to extract keyword from whole document. However, for all sequences, from whole document, allowing arbitrary length, to conduct the above checking method needs enormous computational cost especially the size of the material was large as in this study. Note again our materials were not segmented into words, therefore we must regard arbitrary slit ID as possibly to be starting point or ending point.

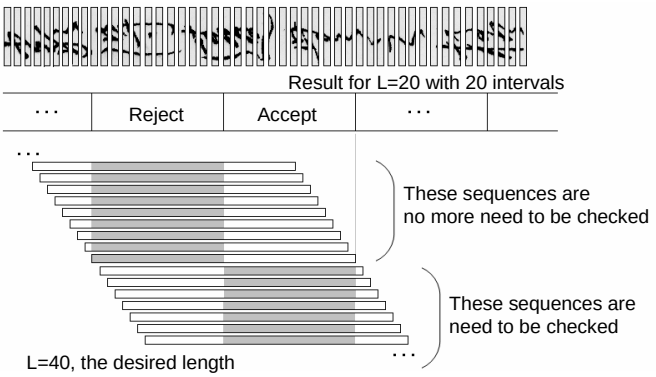


Fig. 5. The pruning method

To reduce the computational cost, pruning was carried out. In pruning, we used the assumption that if a sequence is acceptable as a candidate, the subsequence of it should have been also acceptable. To be exact, this assumption is not perfectly true. However, it is approximately satisfied. Theoretically the average matching cost of subsequence can take a value L/s times larger than the original sequence at extreme case, where L and s are the length of the original sequence and the length of the subsequence, respectively. However, such extreme value rarely appears in practice, which was confirmed in the subsequent experiment.

Under this assumption, now we have to consider only just 40-length slit sequences, because sequences with more than 40-length will be obtained by merger of consecutive 40-length slit sequences.

Therefore let us consider extracting the candidates of keyword with 40-length. The simplest method is to check the average matching cost for all 40-length slit sequences, however it takes too much computational cost. On the other hand, alternative method is to conduct pruning before doing complete search. As a pruning, we checked the average matching cost for 20-length slit sequences at intervals of 20 slit. In this step, about three quarters sequences were rejected although the threshold value was loose than original. After that,40-length slit sequences which includes the accepted 20-length sequences were checked (see Fig. 5). It cost about a quarter of the simplest method, because three quarters sequences were already rejected in the pruning step. We must mention that the computational cost of pruning step was far lower than the subsequent step for two reasons. One is that pruning step scans the document at intervals of 20 slits, therefore the number need to be checked is only 1/20 to original. Another reason is that the computational cost of DTW is quite sensitive to the length of the sequence. Checking with half length is considerably faster than the original length. For these reasons, the computational cost added by the pruning step was far lower than the cost which saved.

Applying above pruning technique recursively, we could reduce computational cost considerably. That is, after 20-length sequences were checked whether

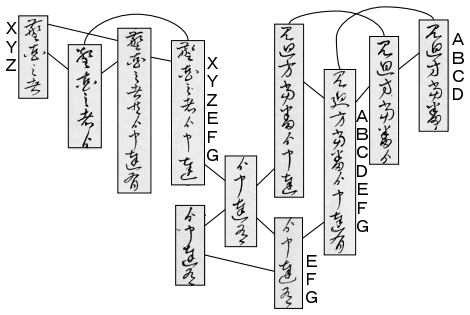


Fig. 6. Failure in clustering

or not being acceptable, then 30-length sequences were checked, and after that, then 35-length were checked, approaching original 40-length. With the use of this technique, computational cost declined to 6% of thorough search, while 96% of the relevant result were retained.

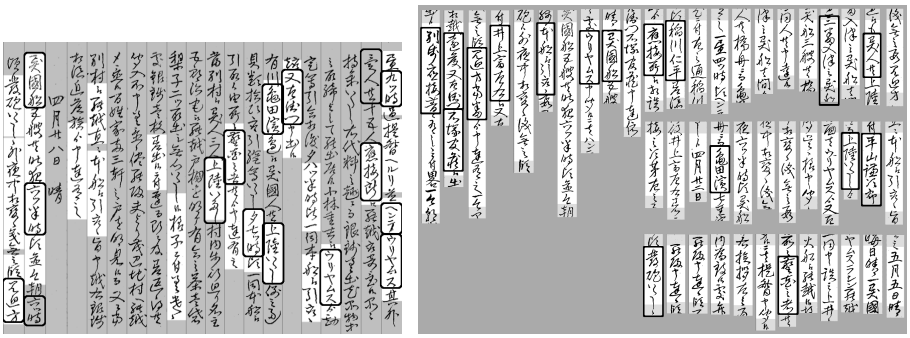
3.3 Clustering Candidate

The candidates of keywords were extracted in the previous method, however they should be large amount and should have quite many duplications. All that remains to do was transforming it into the set of keywords. For this purpose, a clustering method was applied.

In this process, we used graph theory algorithms. For every candidate of keyword, the starting slit ID was regarded as a vertex of the graph. Two vertices were connected if and only if the matching cost between the images starting from corresponding slit ID were under a certain threshold. After this graph was constructed, every connected parts were output as a keyword cluster. In this definition, connected means that there exists a path (not necessarily an edge) between all pairs of vertices in the cluster. The benefit of this definition is that by merging an image and an image with only slight displacement, redundant output can be avoided.

However, simply applying single connection analysis is prone to invite an unwelcome result. It sometimes connects clusters too much. For example, consider clusters C_1 that includes keyword sequence '*abcd*.' If there exists sequences such as '$abcdefg$', these sequences are merged into cluster C_1. Furthermore, if there exists sequences such as '$xyzefg$', these sequences are also merged into C_1 because their subsequence 'efg' are corresponding. By this way, keyword '$abcd$' and 'xyz' are merged into same clusters. It is an unwelcome result.

First attempt to avoid this over-connection was to remove unwanted vertices from graph. Every vertices should have sufficient numbers of adjacent vertices because they were survivors of the keyword candidate extraction. However, there exist some vertices with less numbers of adjacent vertices. Such a less-connected vertices were removed from graph because they were likely to be an error. In practice, we removed vertices which had less than five adjacent vertices.



(a) Selected keyword candidate (white back area)

(b) Final output: Every keyword candidates are clustered

Fig. 7. Result of keyword extraction. Solid box represents manually marked keyword

Another attempt we have made was hierarchical extraction algorithm which could avoid over-connections such as '$abcd$' and 'xyz' described above. First, longer keywords such as '$abcdefg$' and '$xyzefg$' were clustered. Removing such keywords from graph, then shorter keywords were clustered. By this hierarchical extraction, keyword '$abcd$' and 'xyz' might not be extracted, however we consider it is also good because keyword '$abcdefg$' was already extracted and '$abcd$' could be found as subsequence in it.

Finally, representative images were chosen from each clusters. The set of representative images were our final output, which is displayed in next section.

4 Experimental Results

The algorithms presented above were applied to the "Akoku Rasishiki," the detail of which is described in Sect. 1.1.

Figure 7(a) shows the extracted candidates of keywords. In the figure, white-back region represents extracted candidates of keywords, while solid box represents the manually marked keywords, the ground truth. In this figure, among sixteen manually marked ground truths, fourteen were extracted as a candidate correctly. Here we must mention that it is not necessary for every keyword to be extracted as a candidate. Since the keyword will appear in the document several times, if the sufficient number of instances are extracted as a candidate, the system is still able to output it as a keyword.

The final result after candidate clustering is displayed in Fig. 7(b), where 46 keyword images were output. As expected, all of these keyword images were frequently appearing images in the document, including stopwords and meaningless character strings. Among of them, 17 valid keywords were contained among 23 ground truth. Recall rate was 74% and precision rate was 37%. Although precision rate was sacrificed because our method did not use lexical information, high recall rate was worthy of remark.

5 Conclusions and Future Work

In this paper, two main contributions are described. The first half describes the text retrieval method when given a query word. The ranked list sorted by similarity was produced, and it was evaluated in recall-precision manner. The average precision reached over 80–90%. The second half describes the keyword extraction method, which works even if no query is explicitly specified. It is noteworthy that both algorithms works well for Japanese documents, where segmentation into word is practically impossible.

Our future work will focus on considering more sophisticated design for keyword extraction and its clustering. Further experiment, for example applying the method to other languages is also projected, because our method is conceptually independent of languages. Improving the preprocessing for further enhanced performance is also in progress.

References

1. G.A. Fink and T. Plötz "On appearance-based feature extraction methods for writer-independent handwritten text recognition," Proc. of International Conference on Document Analysis and Recognition, pp. 1070–1074, 2005.
2. B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis, and S. Perantonis, "A segmentation-free approach for keyword search in historical typewritten documents," Proc. of International Conference on Document Analysis and Recognition, pp. 54–58, 2005.
3. Yue Lu and Chew Lim Tan, "Word spotting in Chinese document images without layout analysis," Proc. of IEEE International Conference on Pattern Recognition, pp. 30057–30060, 2002.
4. R. Manmatha, Chengfeng Han and E.M. Riseman, "Word Spotting: A New Approach to Indexing Handwriting," Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 631–637, 1996.
5. S. Marinai, E. Marino, and G. Soda, "Indexing and retrieval of words in old documents," Proc. of International Conference on Document Analysis and Recognition, pp. 223–227, 2003.
6. R. Oka, "Spotting Method for Classification of Real World Data," The Computer Journal, vol. 41, no. 8, pp. 559–565, 1998.
7. T.M. Rath and R. Manmatha, "Features for Word Spotting in Historical Manuscripts," Proc. of International Conference on Document Analysis and Recognition, pp. 218–222, 2003.
8. T.M. Rath and R. Manmatha, "Word image matching using dynamic time warping," Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 521–527, 2003.
9. K. Terasawa, T. Nagasaki, and T. Kawashima, "Eigenspace method for text retrieval in historical document images," Proc. of International Conference on Document Analysis and Recognition, pp. 437–441, 2005.
10. M.A. Turk and A.P. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71–86, 1991.
11. M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586–591, 1991.